

We Claim:

1. In a graphical modeling environment having at least one model with a plurality of executable time-based components and which provides a view of the model, a

5 method, comprising the steps of:

monitoring the execution of said model to determine the occurrence of a specified event, said event referenced by a label;

determining the occurrence of said specified event during the execution of said model; and

10 posting the occurrence of said event in said graphical modeling environment to an event handler, said posting notifying said event handler of the occurrence of said event; and

executing at least one component from said plurality of components in response to said notifying, said component associated with said label.

15

2. The method of claim 1, comprising the further steps of:

registering at least one of said plurality of components with said event handler; and

20 receiving at the at least one of said plurality of components registering with said event handler, notification of the occurrence of said event following said posting.

3. The method of claim 1, comprising the further step of:

displaying a post component on said view, said post component specifying a condition causing said posting to occur.

25

4. The method of claim 1, comprising the further step of:

setting a sample time for the initial execution of at least one component to be the occurrence of the specified event,

30 5. The method of claim 4, comprising the further step of:

propagating the sample time to at least one other component in said model, said at least one other component configured to inherit a sample rate.

6. The method of claim 4, comprising the further step of:
 setting a sample time of a plurality of non-contiguous components in said
model to be the occurrence of said event.

5

7. The method of claim 6 wherein said sample time for the plurality of non-
contiguous components is set without adjusting visible connections between
components displayed in said view.

10 8. The method of claim 4, comprising the further step of:
 indicating with an event ID in said view that the sample time of said at least
one component is set to said event.

9. The method of claim 4 wherein said event is an explicit event set by a user.

15

10. The method of claim 4 wherein said event is an implicit event caused by the
execution of the model.

11. The method of claim 10 wherein the implicit event is one of power-up, power-
20 down and initialization.

12. The method of claim 10 wherein the implicit event corresponds to one of the
enabling and disabling of a subsystem.

25 13. The method of claim 2, comprising the further step of:
 indicating which event a component receives with a user-configurable color in
said view.

14. The method of claim 1, wherein an execution scope of the specified event for
30 which the execution of the model is being monitored is restricted to a portion of the
model.

15. The method of claim 1 wherein each event in said model maps on a one-to-one
basis to an event handler, said event handler being a function.

16. The method of claim 15 wherein said function is inlined.

17. The method of claim 1 wherein a branch priority block indicates an order of
5 execution among at least two branches of blocks in response to said notifying.

18. The method of claim 1 wherein more than one block group executes in response
to said notifying, said block groups being a user selected grouping of blocks, the
order of execution of the block groups specified by a user.

10

19. In a modeling environment having at least one model with a plurality of
executable components; a method, comprising the steps of:

monitoring the execution of said model to determine the occurrence of a
specified event;

15 determining the occurrence of said specified event during the execution of said
model;

interrupting execution of an executing event in response to the determination
of the occurrence of said specified event.; and

20 performing an operation in said model in response to the determination of the
occurrence of the specified event.

20. The method of claim 19 wherein said specified event is treated as a normal event
and comprising the further step of:

resuming execution of the interrupted event.

25

21. The method of claim 19 wherein said specified event is treated as an exception
event and comprising the further step of:

returning control of the execution of the model to a calling process which
called the interrupted executing event without resuming execution of said interrupted
30 event.

22. The method of claim 19 wherein said specified event is specified using an
instantiated event object.

23. The method of claim 22 wherein said event is an explicit event.

24. The method of claim 22 wherein said event is an implicit event.

5 25. The method of claim 22 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

26. The method of claim 25 wherein said task object has at least one of a specified execution rate and priority.

10

27. The method of claim 26 wherein at least two events with different tasks are executing in a model and comprising the further step of:

using event transition components to schedule the execution of components associated with said at least two events, said event transition components separating
15 the execution of said components associated with said at least two events.

28. The method of claim 19 wherein the operation is controlled by an order of execution indicated in a branch priority block.

20 29. The method of claim 19 wherein the operation is the execution of more than one block group, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.

30. In a modeling environment, a system, comprising:

25 at least one graphical model with a plurality of executable components;
an event handler, said event handler receiving notice from said model of the occurrence of a specified event; and

at least one receiving component, said receiving block receiving notification from said event handler regarding the occurrence of said specified event and
30 executing in response to said notification.

31. The system of claim 30 wherein said event is an error event stemming from the execution of the model.

32. The system of claim 30 wherein the occurrence of said specified event causes more than one component in said model to execute.

33. In an electronic device a medium holding a graphical modeling environment
5 having at least one model with a plurality of executable time-based components, said graphical modeling environment providing a view of the model, said medium holding executable steps for a method, said method comprising the steps of: /

monitoring the execution of said model to determine the occurrence of a specified event, said event referenced by a label;

10 determining the occurrence of said specified event during the execution of said model;

posting the occurrence of said event in said modeling environment to an event handler, said posting notifying said event handler of the occurrence of said event; and

15 executing at least one component from said plurality of components in response to said notifying, said component associated with said label.

34. The medium of claim 33, wherein said method comprises the further steps of:

registering at least one of said plurality of components with said event handler; and

20 receiving at the at least one of said plurality of components registering with said event handler notification of the occurrence of said event following said posting.

35. The medium of claim 33, wherein said method comprises the further step of:

25 displaying a post component on said view, said post component specifying a condition causing said posting to occur.

36. The medium of claim 33, wherein said method comprises the further step of:

30 setting a sample time for the initial execution of at least one component to be the occurrence of the specified event,

37. The medium of claim 36, wherein said method comprises the further step of:

propagating the sample time to at least one other component in said model, said at least one other component configured to inherit a sample rate.

38. The medium of claim 36, wherein said method comprises the further step of:
 setting a sample time of a plurality of non-contiguous components in said
model to be the occurrence of said event.

5 39. The medium of claim 38 wherein said sample time for the plurality of non-
contiguous components is set without adjusting visible connections between
components displayed in said view.

10 40. The medium of claim 36, wherein said method comprises the further step of:
 indicating with an event ID in said view that the sample time of said at least
one component is set to said event.

41. The medium of claim 36 wherein said event is an explicit event set by a user.

15 42. The medium of claim 36 wherein said event is an implicit event caused by the
execution of the model.

43. The medium of claim 42 wherein the implicit event is one of power-up, power-
down and initialization.

20 44. The medium of claim 42 wherein the implicit event corresponds to one of the
enabling and disabling of a subsystem.

25 45. The medium of claim 34, wherein said method comprises the further step of:
 indicating which event a component receives with a user-configurable color in
said view.

30 46. The medium of claim 33, wherein an execution scope of the specified event for
which the execution of the model is being monitored is restricted to a portion of the
model.

47. The medium of claim 33 wherein each event in said model maps on a one-to-one
basis to an event handler, said event handler being a function.

48. The medium of claim 47 wherein said function is inlined.

49. The medium of claim 33 wherein a branch priority block indicates an order of execution among at least two branches of blocks in response to said notifying.

5

50. The medium of claim 33 wherein more than one block group executes in response to said notifying, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.

10 51. In an electronic device, a medium holding a modeling environment having at least one model with a plurality of executable components, said medium holding executable steps for a method, said method comprising the steps of: /

monitoring the execution of said model to determine the occurrence of a specified event;

15 determining the occurrence of said specified event during the execution of said model;

interrupting execution of an executing event in response to the determination of the occurrence of said specified event.; and

20 performing an operation in said model in response to the determination of the occurrence of the specified event.

52. The medium of claim 51 wherein said specified event is treated as a normal event and wherein said method comprises the further step of:

resuming execution of the interrupted event.

25

53. The medium of claim 51 wherein said specified event is treated as an exception event and comprising the further step of:

30 returning control of the execution of the model to a calling process which called the interrupted executing event without resuming execution of said interrupted event.

54. The medium of claim 51 wherein said specified event is specified using an instantiated event object.

55. The medium of claim 54 wherein said event is an explicit event.

56. The medium of claim 54 wherein said event is an implicit event.

5 57. The medium of claim 54 wherein said event object is associated with a task object, said task object corresponding to an operating system task.

58. The medium of claim 57 wherein said task object has at least one of a specified execution rate and priority.

10

59. The medium of claim 58 wherein at least two events with different tasks are executing in a model and wherein said method comprises the further step of:
using event transition components to schedule the execution of components associated with said at least two events, said event transition components separating
15 the execution of said components associated with said at least two events.

60. The medium of claim 51 wherein the operation is controlled by an order of execution indicated a branch priority block.

20 61. The medium of claim 51 wherein the operation is the execution of more than one block group, said block groups being a user selected grouping of blocks, the order of execution of the block groups specified by a user.